## REMARKS

The Office Action dated March 25, 2005 has been received and carefully noted. The above amendments to the claims, and the following remarks, are submitted as a full and complete response thereto.

Claims 2, 3, 7, 8, 11 and 12 have been amended. No new matter has been added, and no new issues are raised which require further consideration and/or search. Claims 1-13 are submitted for consideration.

Claims 2, 3, 7, 8, 11 and 12 were objected to because of informalities. Claims 2, 3, 7, 8, 11 and 12 have been amended to overcome the objection. Therefore, Applicants request that the objection be withdrawn.

Claims 1, 6 and 10 were rejected under 35 U.S.C. 103(a) as being obvious over U.S. Patent No. 6,122,279 to Milway et al in view of U.S. Patent No. 4,613,954 to Sheth and further in view of European Patent No. 0 572 145 A2 to Thompson et al. The rejection is traversed as being based on references that neither teaches nor suggests the novel combination of features clearly recited in independent claims 1, 6 and 10.

Claim 1, upon which claims 2-5 depend, recites a network device that is configured to prevent data misalignment of a data packet containing extra header bytes. The network device includes an ingress module having an input interface to receive a cell of the data packet and a header detector configured to detect a header of a cell of the data packet and remove the header from the cell of the data packet. The network device also

includes a counter to determine whether the cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed. The network device further includes an insertion module configured to insert null bytes into the cell of the data packet to form a modified cell of the data packet if the counter determines that the cell of the data packet does not satisfy the multiple of the predetermined number of bytes. The network device also includes an extraction module configured to remove the null bytes from the modified cell of the data packet as the modified cell of the data packet exits the network device.

Claim 6, upon which claims 7-9 depend, recites a method of preventing data misalignment of a data packet containing extra header bytes. The method includes receiving a cell of the data packet at an input port of a network device and detecting a header of a cell of the data packet. The method also includes removing the header from the cell of the data packet and determining whether the cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed. The method further includes inserting null bytes into the cell of the data packet to form a modified cell of the data packet if the counter determines that the cell of the data packet does not satisfy the multiple of the predetermined number of bytes and forwarding the modified cell of the data packet to an output port. The method also includes removing the null bytes from the modified cell of the data packet as the modified cell of the data packet exits the network device.

Claim 10, upon which claims 11-13 depend, recites a network device configured to prevent data misalignment of a data packet containing extra header bytes. The network device includes receiving means for receiving a cell of the data packet at an input port of the network device and detecting means for detecting a header of a cell of the data packet. The network device also includes header removing means for removing the header from the cell of the data packet and determining means for determining whether the cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed. The network device further includes inserting means for inserting null bytes into the packet to form a modified cell of the data packet if the counter determines that the cell of the data packet does not satisfy the multiple of the predetermined number of bytes and forwarding means for forwarding the modified cell of the data packet to an output port. The network device also include null byte removing means for removing the null bytes from the modified cell of the data packet as the modified cell of the data packet exits the network device.

As will be discussed below, the cited prior art references of Milway, Sheth and Thompson et al. fail to disclose or suggest the elements of any of the presently pending claims.

Milway teaches an ATM switch with a microprocessor, a switch controller, a memory, a token grant logic and port clusters, wherein each port cluster contains line interfaces, port logic and buffering for up to eight ATM network connections. Col. 6, lines 27-33. Cell data is delivered from one cluster to another by a switch bus. Col. 7,

lines 3-4. The principal task of the switch is to route ATM cells from a plurality of input links to a plurality of output links. Col. 7, lines 22-24. In operation, ATM cells are received by the switch via an ATM line interface. A new ATM cell arriving on an input link is converted from electrical signals to a bit stream that is provided to a network control logic which checks for errors and discards misdirected cells. Col. 9, lines 1-11.

Sheth teaches a data transfer network wherein a peripheral-controller is used to manage and control data transfer operations between a peripheral and a main host computer system, whereby data is transferred rapidly in large blocks. Col. 2, lines 43-49. The host system, to initiate an operation, sends to the peripheral controller an I/O descriptor that specifies the operation to be performed and descriptor link words that include path selection information and identify the task to be performed. Upon receiving the information the peripheral controller makes a transition to a state. When the operation is completed, the peripheral controller returns a result descriptor indicating the status of the operation to the host system. Col. 4, lines 10-47.

The system has capabilities of using a burst mode wherein data can be transferred to the host system at 64 megabits per second. Col. 11, lines 13-21. When in the burst mode, a burst counter maintains a count of the number of words remaining to be transferred between the host and the peripheral controller during the burst transfer cycle. In the normal situation when there are two or more blocks of data to be transferred to the host system, the controller sets the burst counter to 256 words and sends blocks of data to the host in the burst mode. When there are less than two blocks of data remaining to

complete the I/O operation, the controller calculates the actual length of the remaining data by comparing the P register and the S register. If the remaining number is odd, the final byte is the PAD byte and all zeros are inserted by the peripheral controller. The final two blocks are sent to the host by the controller on a word by word transfer basis, wherein each word is transferred individually rather than automatically as in the burst mode. Col. 13, lines 19-34.

Thompson teaches a computer system with a processor, a cache, a memory and a network adapter. The network adapter generates and inserts network data checksums. In the outbound direction, the processor provides checksum control information to the network adapter and the network adapter calculates the checksum and inserts the checksum into the proper location within the packet before transmitting the packet on the network. In the inbound direction, the network adapter decodes the packet header, programs the checksum control information directly into internal registers, calculates the checksum and inserts the checksum into the proper location within the packet before transmitting the packet on the memory. The network adapter also automatically separates headers and data during transfer of incoming packets from the adapter to the memory. The network data further performs alignment of network headers by inserting pad bytes based on specific values found in the network link header. Col. 3, line 1-Col 4, line 50.

The network adapter is connected to the network through a front plane controller that provides transmission and reception of data packets to and from the network. For outbound transfers, the front plane controller unpacks the words from a DMA bus, looks

at the first byte of the output stream, which contains a count of how many pad bytes were inserted in the packet and strips off the pad bytes. Col. 6, lines 35-46.

Applicants submit that the combination of Milway, Sheth and Thompson et al., simply does not teach or suggest the combination of features clearly recited in claims 1, 6 and 10. Claims 1, 6 and 10, in part, recite a counter for determining whether the cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed and an insertion module configured to insert null bytes into the cell of the data packet to form a modified cell of the data packet if the counter determines that the cell of the data packet does not satisfy the multiple of the predetermined number of bytes. The Office Action correctly notes that Milway does not teach or suggest the counter and the insertion module recited in claims 1, 6 and 10. However, the Office Action states that both Sheth and Thompson et al. teach the counter and the insertion module recited in claims 1, 6 and 10.

As mentioned above, Sheth teaches a burst counter that maintains a count of the number of words remaining to be transferred between the host and the controller during the burst transfer cycle and when there are less than two blocks of data remaining to complete the I/O operation, the controller calculates the actual length of the remaining data and if the remaining number is odd, the final byte is the PAD byte and all zeros are inserted by the control. There simply is no teaching or suggestion in Sheth of a counter for determining whether the cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed as recited in claims 1, 6 and 10. The

Office Action also correctly notes that Sheth does not teach or suggest the insertion module recited in claims 1, 6 and 10.

Thompson et al. simply does not cure the deficiencies of Sheth and Milway. There is simply no teaching or suggestion in Thompson of a counter for determining whether the cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed as recited in claims 1, 6 and 10. Thompson et al. teaches that a network adapter calculates a checksum either based on control information placed in the packet by the processor or based on checksum control information that is programmed into internal registers. According to Thompson, when the packet is transmitted from the network adapter to memory, the processor compares the checksum result inserted into the packet by the network adapter against the packet checksum to verify the data. See. Col. 3, line 49- Col. 4, line 13. Furthermore, contrary to what is stated in the Office Action, Thompson et al. does not teach or suggest an insertion module configured to insert null bytes into the cell of the data packet to form a modified cell of the data packet if the counter determines that the cell of the data packet does not satisfy the multiple of the predetermined number of bytes as recited in claims 1, 6 and 10. Col. 6, lines 35-46 of Thompson et al. does teach for outbound transfers, the front plane controller unpacks the words from a DMA bus, looks at the first byte of the output stream, which contains a count of how many pad bytes were inserted in the packet and strips off the pad bytes. However, there is no teaching of suggestion in Thompson that the front plane controller inserts null bytes into the cell of the data packet to form a

modified cell of the data packet <u>if the counter determines that the cell of the data packet does not satisfy the multiple of the predetermined number of bytes</u> as recited in claims 1, 6 and 10. Therefore, Applicants respectfully assert that the rejection under 35 U.S.C. §103(a) should be withdrawn because neither Milway, Sheth nor Thompson et al., whether taken singly or combined, teaches or suggests each feature of claims 1, 6 and 10.

Claims 2-4, 7-8 and 11-12 were rejected under 35 U.S.C. 103(a) as being obvious over U.S. Milway et al in view of Sheth and Thompson et al. and further in view of U.S. Patent No. 6,567,413 B1 to Denton. The rejection is traversed as being based on references that neither teaches nor suggests the novel combination of features clearly recited in independent claims 1, 6 and 10.

Denton also does not cure the deficiencies of Sheth, Milway and/or Thompson et al. Denton teaches an optical networking module that is formed with an integrated module including optical, optical-electrical and protocol processing components and complementary software. There is simply no teaching or suggestion in Denton of a counter for determining whether the cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed and an insertion module configured to insert null bytes into the cell of the data packet to form a modified cell of the data packet if the counter determines that the cell of the data packet does not satisfy the multiple of the predetermined number of bytes as recited in claims 1, 6 and 10. Therefore, Applicants respectfully assert that the rejection under 35 U.S.C. §103(a) should be withdrawn because neither Milway, Sheth, Thompson et al. nor Denton,

whether taken singly or combined, teaches or suggests each feature of claims 1, 6 and 10 and hence dependent claims 2-4, 7-8 and 11-12, thereon.

Claims 5, 9 and 13 were rejected under 35 U.S.C. 103(a) as being obvious over U.S. Milway et al in view of Sheth and Thompson et al. and further in view of U.S. Patent No. 6,697,873 B1 to Yik et al. The rejection is traversed as being based on references that neither teaches nor suggests the novel combination of features clearly recited in independent claims 1, 6 and 10.

Yik et al. also does not cure the deficiencies of Sheth, Milway and/or Thompson et al. Yik et al teaches an apparatus and method for storing and searching computer node addresses in a computer network system. There is simply no teaching or suggestion in Denton of a counter for determining whether the cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed and an insertion module configured to insert null bytes into the cell of the data packet to form a modified cell of the data packet if the counter determines that the cell of the data packet does not satisfy the multiple of the predetermined number of bytes as recited in claims 1, 6 and 10. Therefore, Applicants respectfully assert that the rejection under 35 U.S.C. §103(a) should be withdrawn because neither Milway, Sheth, Thompson et al. nor Yik et al., whether taken singly or combined, teaches or suggests each feature of claims 1, 6 and 10 and hence dependent claims 5, 9 and 13, thereon.

As noted previously, claims 1-13 recite subject matter which is neither disclosed nor suggested in the prior art references cited in the Office Action. It is therefore

respectfully requested that all of claims 1-13 be allowed and this application passed to issue.

If for any reason the Examiner determines that the application is not now in condition for allowance, it is respectfully requested that the Examiner contact, by telephone, the applicants' undersigned attorney at the indicated telephone number to arrange for an interview to expedite the disposition of this application.

In the event this paper is not being timely filed, the applicant respectfully petitions for an appropriate extension of time. Any fees for such an extension together with any additional fees may be charged to Counsel's Deposit Account 50-2222.

Respectfully submitted,

Arlene P. Neal

Arlene P. Neal
Registration No. 43,828

**Customer No. 32294**
SQUIRE, SANDERS & DEMPSEY LLP
14TH Floor
8000 Towers Crescent Drive
Tysons Corner, Virginia 22182-2700
Telephone: 703-720-7800
Fax: 703-720-7802

APN:mm